

## AERODYNAMIC SHAPE OPTIMIZATION USING UNSTRUCTURED GRID METHODS

Susan E. Cliff, Senior Member, Aerospace Engineer, NASA/Ames, Moffett Field, CA  
Scott D. Thomas, Senior Member, Programmer/Analyst, Raytheon, Moffett Field, CA  
Timothy J. Baker, Senior Research Scientist, Princeton University, Princeton, NJ  
Antony Jameson, AIAA Fellow, Professor of Engineering, Stanford University, Stanford, CA  
Raymond M. Hicks, Retired, Wickenburg, AZ

### ABSTRACT

Two unstructured-grid Euler-based CFD codes, AIRPLANE and FLOWCART were coupled to a gradient-based quasi-Newton finite-difference optimization algorithm. These two optimization techniques were developed to provide detailed aerodynamic shape optimization methods for complete configurations with efficient grid generation methods. AIRPLANE utilizes a tetrahedral mesh whereas FLOWCART uses a hexahedral Cartesian mesh. Several codes were developed to facilitate aerodynamic shape optimization. These include developing surface grid perturbation methods with thickness constraint and overall surface overlap evaluations, used with both Euler codes, and adding a multi-grid capability and a variety of mesh movement techniques to the AIRPLANE method. The AIRPLANE multigridding approach was proven to be accurate and effective, with typical speedup ratios of 3 to 5. The mesh movement techniques were effective in reducing the grid generation wall clock time by 70%. Detailed results of the AIRPLANE-based optimization technique are presented. The performance gains resulting from optimization are verified by computations with FLOWCART and OVERFLOW, and comparisons with experimental data on the baseline and optimized configurations. The low speed computational results of the baseline and optimized models were incorporated into an approach and landing simulation database. The controllability and handling qualities were good to excellent based on a piloted simulation in the NASA Ames Vertical Motion Simulator (VMS). FLOWCART-based optimization was validated by comparing its gradients and design solutions with AIRPLANE's on two separate optimization problems with identical design variables and objective functions.

### INTRODUCTION

Aerodynamic shape optimization using structured grids was used successfully for several years at NASA Ames on High Speed Civil Transport designs.<sup>1</sup> An adjoint method enabled rapid gradient computation,<sup>2,4</sup> and efficient surface and volume mesh movement techniques<sup>5</sup> were applied. However, these structured grid methods required start-up periods of a few months of labor to generate suitable meshes for complete configurations.

The volume meshes consisted of many abutting zones, with points matching at abutting boundaries. For multigridding, each coarser mesh required half the number of points in each direction or one-eighth of the total grid points of the next finer mesh, matching alternate points in each direction. In addition to the full configuration mesh, another surface representation, consisting of sectional cuts, was required for rapid surface perturbations during optimization.

Unstructured grid methods can establish the baseline surface and volume grid in far less time and more easily than for structured grid methods. Volume grid generation is completely automated for both Cartesian and tetrahedral approaches, independent of the geometric complexity of the configuration. These methods allow point insertion and deletion without introducing hanging nodes or extraneous points. Thus, the unstructured approach is well suited for the emergence or removal of surface components during optimization, or for adaptation to dynamic changes in flow field phenomena.

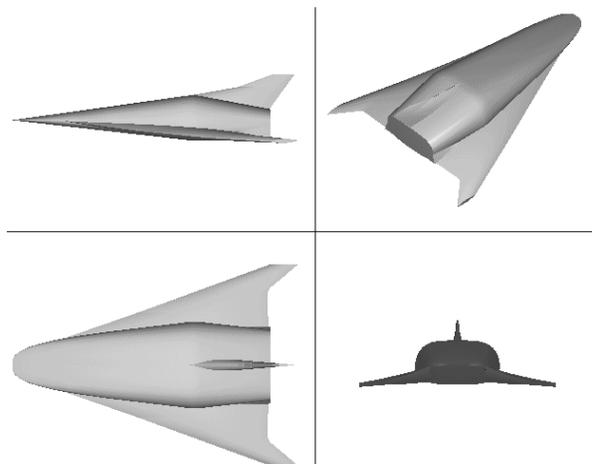


Fig. 1. Baseline CTV Configuration.

Complete configuration multipoint numerical optimization can begin within a few days using unstructured methods. Optimization results can be obtained within a couple of weeks with a dedicated computer system, potentially before a structured-grid based optimization program could even begin. The negative side to using an unstructured optimization technique is that it is more expensive computationally and requires more memory. In addition, there is not currently an adjoint solver available, and the expense of finite-difference gradient computations severely limits the number of design variables that can be used simultaneously. In spite of these limitations, the reduced grid generation times for the unstructured methods may still tip the balance in their favor.

Examples of the AIRPLANE based optimization process will be illustrated through the redesign of a Crew Transfer Vehicle (CTV) at multiple design points. NASA Ames Research Center personnel are conducting conceptual design studies of sharp-edged re-entry vehicles.<sup>6</sup> The leading edges of the nose and wing will be composed of a new Ultra High Temperature Ceramic (UHTC) material<sup>7</sup> to withstand the re-entry heat loads. These vehicles are being evaluated since they may improve the re-entry cross range capability and therefore the safety during abort mission scenarios. The planform, wing twist and dihedral have recently been modified since the publication of the original baseline<sup>6</sup> to improve the low speed trim characteristics of the model. The modified baseline CTV is shown in Fig. 1.

The stability and control of this class of vehicle is problematic since the vehicle must operate from hypersonic speeds during re-entry to subsonic speeds upon approach and landing, and trim must be achieved throughout this extensive Mach number range. AIRPLANE was used to conduct detailed simultaneous optimization at the landing speed of Mach 0.3, and the re-entry descent speed of Mach 6.0. The design objective was to trim the vehicle at the two conditions simultaneously while achieving the best L/D in the process. The results for the baseline and optimized CTVs from AIRPLANE based optimizations are compared with FLOWCART, OVERFLOW Navier-Stokes analyses, and experimental results from a low-speed wind tunnel. In addition NASA astronaut piloted time-histories from VMS flight simulation data for the baseline and optimized configurations for approach and landing will be presented.

#### **NUMERICAL OPTIMIZATION**

A gradient method was employed since it is well suited to detailed aerodynamic design. Gradient-based algorithms such as QNMDIF<sup>8,9</sup> should require fewer function evaluations to find local minima or design improvements in contrast to non-gradient methods such as

genetic algorithm (GA) techniques. QNMDIF is an unconstrained quasi-Newton finite-difference method that is capable of achieving a local minimum in a design space, but it is not guaranteed to find the true global minimum. The gradient calculations are somewhat expensive since they require two flow solutions for each design variable for forward and backward difference calculations. An AIRPLANE adjoint method is currently under development using techniques that have been successfully developed for aerodynamic design.<sup>10-14</sup> When this adjoint method is available it will reduce the CPU time for gradient computations to the equivalent of two flow solutions for any number of design variables.

For the present methods, a limited number of design variables is recommended, unless many processors are available. QNMDIF is an unconstrained method with a single objective function, so constraints must be implemented by adding penalty terms to the objective function.

QNMDIF calls a user-supplied function named FUN that returns the value of the objective for a prescribed set of design variables. An overview of the function FUN using AIRPLANE-based optimization is shown in Fig. 2. FUN controls surface shape perturbation, mesh movement or grid regeneration, and computation of the flow solution. This function is called repeatedly during the gradient and search steps of the optimization. The optimization code stops after a user-supplied number of design iterations is reached or when no further improvements can be made for the current design variables.

The user supplies a range of values for each design variable to constrain the sizes of the perturbations. If the variables are outside of this range, a penalty function is returned to QNMDIF. If the design variables are in range, the code checks for a stored solution/objective function, matching the flow conditions and design variables. This is used to “restart” the optimization process by quickly retracing the steps of previous optimization runs. If no existing solution/objective function is found, the code will proceed to modify the surfaces of the configurations. APShAPER perturbs the triangulated surfaces ensuring that the number of points and connectivity remain unchanged. It then computes surface vertical positions and thicknesses at user-prescribed points and writes the information to a file. The AIRPLANE<sup>15,16</sup> volume mesh of tetrahedra is deformed rather than regenerated using MESHMV. This code also preserves the number of points and connectivity of the original mesh. If the method should invert any tetrahedra, the entire mesh is automatically regenerated using MESH3D.<sup>17</sup> Following this, the multigrid version of AIRPLANE is run. Then the surface vertical

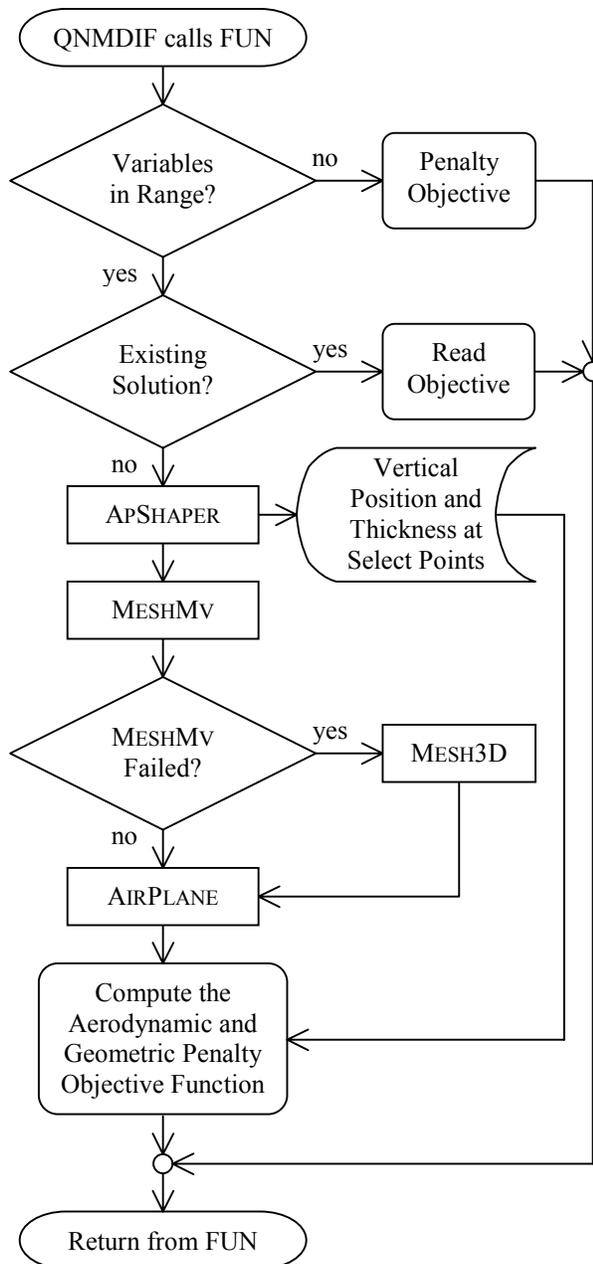


Fig. 2. FUN: Objective Function for QNMDIF.

positions and thicknesses at constraint points previously computed via APShAPER are used to determine the thickness or vertical position differences between baseline and optimized configurations. Penalty terms are nonzero when geometric constraints such as thickness or vertical location are violated. An aerodynamic objective function, such as  $C_D/C_L$ , is added to the geometric penalty constraint terms and returned to QNMDIF. The flowchart for FLOWCART<sup>18,19</sup> is similar to the AIRPLANE flowchart but the mesh is regenerated from scratch rather than deformed during optimization.

### SURFACE MESH PERTURBATION TECHNIQUES: WING AND BODY DESIGN VARIABLES

The triangulated surfaces of wing type components are perturbed during the course of either AIRPLANE-based or FLOWCART-based optimization using the program APShAPER (AP referring to AIRPLANE data structures). The design variables are multipliers of user-defined sets of analytic shape functions added to the wing surfaces. The optimizer adjusts these to improve an objective function. The function minimized is typically a specified aerodynamic quantity such as drag/lift or drag coefficient at fixed lift, but can include other aerodynamic quantities of interest in the form of appropriately weighted penalty function terms, such as a squared difference of the current and target pitching moment coefficients. A few trial and error runs are needed to determine the appropriate weighting parameters.

The most commonly used shape functions are sine bumps that reach a peak value of unity at a user-defined chordwise position and vanish at the wing leading and trailing edges. The designer chooses the rate at which each shape function decays to zero via the function's "width" exponent. The sine bumps are typically centered at uniformly spaced intervals along the chord at wing defining stations. These rows of design variables are lofted over user specified portions of the span. Other wing design variables include polynomial functions for leading- and trailing-edge droop, exponential functions to modify the leading-edge bluntness, twist, and longitudinal and vertical displacement functions to modify the planform or dihedral of the wing, respectively. Use of shape functions allows for isolated regions of the configurations to be modified, reducing the number of design variables for the design problem.

APShAPER requires a surface (U-V) parameterization of the wing upper and lower surfaces so that each mesh point maintains an association with the normalized local chord (U) and local span (V) of the wing after surface perturbations. This is accomplished via the program APPLANFM, given the wing leading and trailing edges.

APShAPER computes mesh displacements using the points of a regular (U-V) structured grid covering the wing planform. The structured grid points are perturbed directly, then bilinear mapping is used to perturb each point of the triangulated surface mesh. The perturbation and mapping operations are quick since the grid is regular and the association of mesh points to grid quadrilaterals is explicit.

Body type component perturbations are achieved using APShBODY. The body design variables allow for fore- and aft-body droop, body thickness and body camber changes. Sine bumps can be applied vertically,

spanwise or radially. Thus, the fuselage height and width can be independently perturbed, or the perturbation of both height and width can be performed using radial perturbations, which place the peak value of the sine function at user-specified circumferential locations from the bottom to the top of the fuselage. The bumps can extend beyond the top and bottom of the fuselage to permit the centers of the bumps to be placed at the top or bottom of the body.

APSHBODY differs from APSHAPER in that the body does not require an U-V mapping file, but rather computes the relative longitudinal and circumferential location of each grid point on the fly.

Several design variable perturbations on the wing and fuselage of a CTV are demonstrated in Fig. 3. The left side (pilot's perspective) of the configuration in the figure is prior to perturbation, and the right side is after perturbation. The forebody is drooped and a radial bump extending beyond the crest was used to perturb the centerline of the fuselage to represent a canopy. A radial sine bump near the aft body is shown with its peak value placed approximately half way between the body center and the crown. The forebody and aft body thickness design variables diminish to zero in the axial direction at the curvature break from the sharp to rounded fuselage. The wing dihedral is reduced, and the outboard wing is washed out with  $15^\circ$  of twist. A sine bump was applied to both upper and lower wing surfaces on the wing tip to demonstrate a large camber change. A flap hinged at 20% chord (negative deflection) is simulated on the inboard panel of the wing. Simulations of leading and/or trailing edge flaps about arbitrary hinge lines can be used, and gaps are permitted.

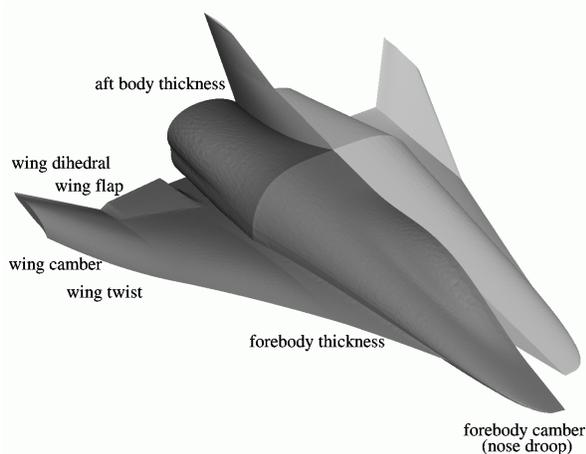


Fig. 3. Example of Surface Shape Functions Applied to the CTV Configuration.

### VOLUME MESH MOVEMENT STRATEGIES

During the course of optimization the volume mesh of tetrahedra is perturbed in response to surface mesh perturbation, rather than being regenerated, if possible. Three reasons for this approach are (1) that it can take less wall clock time to derive a perturbed mesh, (2) the mesh-to-mesh interpolation scheme runs faster by reusing previous results when mesh topologies are preserved, and (3) the gradients are typically more accurate when mesh movement is used compared with mesh regeneration. Further information on mesh movement is available.<sup>20,21</sup>

The idea behind mesh movement is to relax the volume mesh points so that they assume new positions that are compatible with the new boundary position. Three partial differential equation approaches were implemented: Laplace, biharmonic, and stress equilibrium. All of these iterative techniques maintain the mesh topology, and are successful, at convergence, when the mesh is deformed without cell inversion (caused by overlapping edges). It is possible to interpret the Laplace equation approach as modeling the mesh by a set of springs connecting the mesh points along the mesh edges. Thus, the solution of Laplace's equation corresponds to the equilibrium position of a set of springs in response to the new boundary position.

The biharmonic equation can be thought of as a double application of Laplace's equation, and this has the advantage of allowing a greater degree of mesh smoothing. Laplace's equation and the biharmonic equation only allow for a weak coupling between the X, Y, and Z components of displacement.

The equation for stress equilibrium allows X, Y, and Z displacements to be strongly coupled, unlike the previous two methods. In this case, the mesh volume can be thought of as an elastic solid that relaxes to a new position in response to the new boundary position.

Solution of Laplace's equation is the least expensive option; solving the biharmonic equation is more computationally expensive, and solving equations of stress equilibrium involves the largest computational cost. All mesh movement schemes were parallelized to make use of the allotted number of processors during optimization. The loops were restructured, employing OpenMP™ directives.

The three perturbation methods do require more CPU time than mesh regeneration from scratch, if a single processor is used. The mesh movement techniques are more cost effective than re-creating a volume mesh when several CPUs (about 8 or more) are used simultaneously. MESHMV is typically run on 16 processors so that simultaneous movements on successively

coarser meshes run together. This takes about 2 minutes wall time for a 320,129-point mesh with 16 CPUs.

It had been expected that the biharmonic and stress equilibrium equations would permit larger mesh deformations without cell inversion. Experience has shown only marginal improvements over the Laplace method with greater computational cost. The mesh movement techniques work best for small surface perturbations. If cell inversion occurs, the mesh is regenerated by MESH3D. MESH3D is extremely robust and has not been known to fail during optimization. However, the gradient computations have been shown to be less accurate when the mesh is regenerated. This is understood by realizing that mesh regeneration does not guarantee that a consistent number of volume mesh points or that mesh connectivity will be maintained under surface perturbations. Changes in the number of mesh points or connectivity will lead to very small changes in the discrete flow solution which will in turn reduce the accuracy of the gradient calculations.

MESH3D takes about 7 minutes for a 320,129-point grid (1.9 million cells). The Cartesian grid generation method used with FLOWCART takes about 2 minutes for a 1.7 million-point grid. It also has never failed during the course of optimization. Both MESH3D and FLOWCART are scalar, each using only one CPU.

#### MULTIGRID MODIFICATIONS TO AIRPLANE

Multigridding involves the use of two or more unstructured meshes to speed up the transfer of information through the flow field, thus shortening the time needed to find a steady solution.<sup>22,23</sup>

The multigrid Runge-Kutta time stepping scheme requires extra work per cycle. Multigridding pays off because information travels farther on coarser meshes each time step than it can on the finest mesh. Computer time spent on the coarser meshes and transferring information between meshes is a fraction of the time required for the finest mesh. Each coarser mesh typically has 2 to 20 times fewer points than its successor.

Each mesh in the collection is based on an AIRPLANE mesh that could be used by itself, without multigridding. The aircraft surface triangulation should be similar from one mesh to the next, but points do not have to overlap. A few aircraft surface points may be located slightly outside the next coarser or finer mesh, without harm. Triangles on the symmetry plane and far field boundaries are typically on the same Cartesian boundaries. All meshes are stored in memory at the same time, ordered coarse to fine, together with interpolation arrays, communication arrays, metrics, and many other quantities.

#### INTERPOLATION

Information is transferred from one mesh to the other by way of trilinear interpolation. When a point,  $P$ , of one mesh lies geometrically inside a tetrahedral cell,  $ABCD$ , of another mesh, this information is stored in the flow solver as the four addresses of the containing cell vertices together with their four corresponding interpolation coefficients. For example, values of a function  $W$  at each of the cell vertices are combined in the following expression to find an interpolated value,  $WP$ , at point  $P$ :

$$WP = \Phi_A W(A) + \Phi_B W(B) + \Phi_C W(C) + \Phi_D W(D)$$

The interpolation coefficients  $\Phi_A$ ,  $\Phi_B$ ,  $\Phi_C$ ,  $\Phi_D$  are formed by volume ratios, using the opposite face to form the numerator. For example:

$$\Phi_A = (\text{Volume of } PBCD) / (\text{Volume of } ABCD)$$

The four coefficients sum to 1.0, so there are only three independent values.

#### LOST POINTS

Not every point in one mesh can be found in the next coarser or finer mesh. This condition commonly occurs on the aircraft surface. Fig. 4 illustrates this in two dimensions.

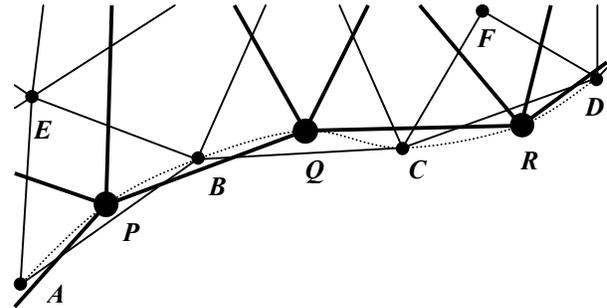


Fig. 4. Lost Points on the Aircraft Surface.

A fine mesh on points  $A$ ,  $B$ ,  $C$ ,  $D$ ,  $E$ , and  $F$  is drawn with thin lines, and a coarse mesh on points  $P$ ,  $Q$ , and  $R$  is drawn with thick lines. The aircraft surface is below the dotted line, which joins points  $A$ ,  $P$ ,  $B$ ,  $Q$ ,  $C$ ,  $R$ , and  $D$ . Both polygonal curves  $ABCD$  and  $PQR$  are on the aircraft surface. Coarse mesh point  $P$  is found in fine mesh triangle  $ABE$ , but coarse mesh point  $R$  is not contained in the fine mesh—it is lost. Similarly, fine mesh points  $A$  and  $B$  are found in the coarse mesh, while  $C$  and  $D$  are lost to the coarse mesh.

In fine-to-coarse interpolation, lost coarse points are assigned values of neighboring points. In the example above, values at fine mesh point  $C$  would be assigned to coarse mesh point  $Q$  because  $C$  is the fine mesh point closest to  $Q$ . This rule is routinely applied to the solution when multigridding.

In coarse-to-fine interpolation, values at lost fine points are set to zero for a correction or to free-stream for a solution. The latter case is not part of multigridding, but is only used when sequencing a converged solution from a coarse mesh to the next finer mesh. The coarse-to-fine interpolation coefficients are applied to aggregate fine mesh corrections to the next coarser mesh, where they are called residuals, and they are also used to interpolate corrections in the other direction, from a coarse mesh to a fine mesh.

### TRANSFER OPERATORS

Information passes between meshes during multigridding by way of three transfer operators:

1. Solution variables are interpolated from fine to coarse by fine-to-coarse coefficients.
2. Residuals are aggregated from fine to coarse by coarse-to-fine coefficients.
3. Corrections are interpolated from coarse to fine by coarse-to-fine coefficients.

The first and third interpolation operators are applied as previously described. The second operator uses the same interpolation coefficients as the third operator to accumulate fine corrections to coarse residuals. The sum of fine corrections equals the sum of coarse residuals, if there are no lost points.

Residual aggregation with cells in 3D is analogous to triangles in 2D, shown in Fig. 5. Thick coarse-mesh edges are drawn over thin fine-mesh edges. Coarse point  $P$  is contained in the pentagonal neighborhood  $ABCDE$ . Each fine mesh point sends portions of its correction to the vertices of the triangle containing it

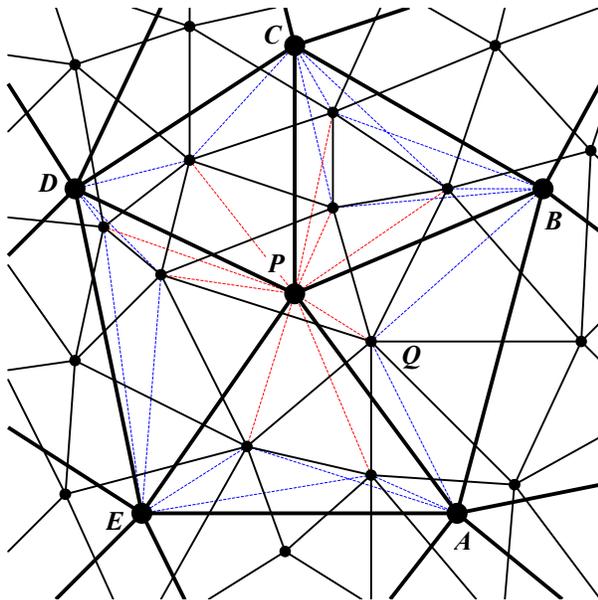


Fig. 5. Aggregate Transfer Operator.

(red and blue dashed lines). The size of the fraction is determined by an area ratio. The fraction of the correction at  $Q$ , for example, that is added to the residual at  $P$  is the area of  $QAB$  divided by the area of  $PAB$ . Only corrections at fine mesh points inside the pentagon contribute to the residual at  $P$  (red dashed lines).

### V AND W SCHEMES

When using two meshes for multigridding, the usual procedure is to alternate work between meshes, in what is known as a V cycle. This is illustrated in the top diagram of Fig. 6. The letter E in each circle indicates

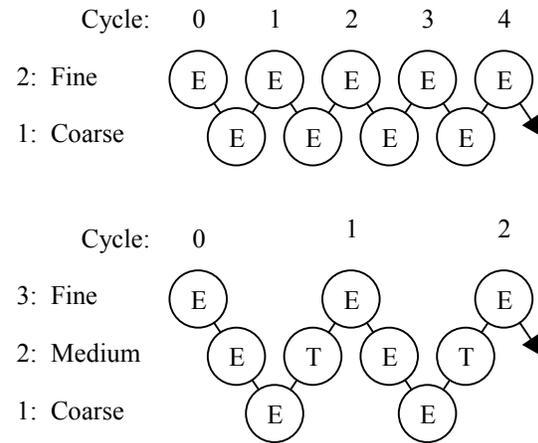


Fig. 6. The V Scheme on 2 and 3 Meshes.

one step in the evaluation of the flow field. In the present case, three meshes are used, which we may call C, M, and F, for Coarse, Medium, and Fine, respectively. The V cycle on three meshes involves working in the order F-M-C-M-F. This is shown in the bottom diagram of Fig. 6. The letter E in each circle has the same meaning as before, and the letter T indicates a transfer of data without updating the solution. After the first F, a cycle in the time stepping scheme consists of -M-C-M-F. Convergence is evaluated at the end of each cycle, and, when the scheme stops, it is the solution on the finest mesh that is used to compute forces.

A more effective way to spread out the work is to repeat the V at the medium level, creating what is known as a W cycle. In this approach, each cycle past the first F consists of -M-C-M-C-M-F. This is illustrated in Fig. 7. The wall time required per cycle increases, but the number of cycles required for convergence is typically reduced, as is the total wall time for convergence.

Many options exist beyond this, especially when using more meshes. For example, adding a third V in the middle of a 3-mesh scheme can improve convergence. It can be useful to move a converged solution from one mesh to the next finer mesh, and then continue multigridding, in what is known as mesh sequencing. When

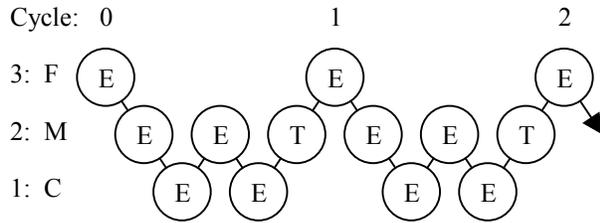


Fig. 7. Two Cycles of The W Scheme on 3 Meshes.

using four or more meshes, a natural approach is to apply the W scheme recursively, see Fig. 8.

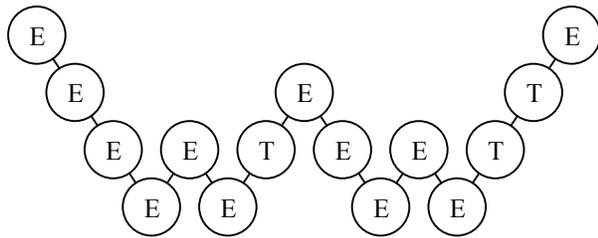


Fig. 8. One Cycle of the W Scheme on 4 Meshes.

Using a W cycle on three meshes is a robust approach, and it is the only variant applied in this report.

The AIRPLANE multigrid flow solver has six free parameters to control aspects of each multigrid cycle. The Courant number, dissipative coefficient, and enthalpy-damping factor may be different for the coarse meshes than for the fine mesh. The far field boundary condition on the coarse meshes may be frozen or updated. A relaxation factor may be applied to the aggregate residuals, and a smoothing operator may be applied to the interpolated corrections. These parameters, once established, may be left alone. The solver is observed to be robust and well suited to its role in the function evaluation part of the optimization process.

#### **THE AIRPLANE MULTIGRIDGING PROCESS**

This section describes the steps involved in running AIRPLANE using three-mesh multigridging in the context of optimization. Coarse, medium, and fine mesh files (numbered 1, 2, and 3, respectively) are created by MESH3D or MESHMV. The three meshes may be generated simultaneously.

The next step is to compute interpolation files using a program named APREPHIS. The name indicates that it is an AIRPLANE (AP) utility that may re-compute (RE) the interpolation coefficients (PHIS). APREPHIS uses multiple processors via compiler directives, and four copies may execute simultaneously. When using three mesh files, there are four mesh pairings that require interpolation coefficients, namely (1,2), (2,1), (2,3), and (3,2). The interpolation files are named

*phis<sub>nm</sub>.unf*, where (*n,m*) indicates the pairing between meshes.

A divide-and-conquer approach keeps the search down to roughly order  $N \log(N)$  operations. When MESHMV is successful, earlier interpolation files may be available, which can further cut the search time by roughly two orders of magnitude. This works because MESHMV preserves topology. APREPHIS checks for a match using old information before searching for a cell in one mesh containing a point in another mesh.

The mesh files are quickly combined together into a file, named *meshmg.unf*, with a program named AP-CATMSH, and the entire domain is split into regions of approximately equal size by PREMG. PREMG is a recursive bisection preprocessor (PRE) that handles multigridging (MG). At this point the number of processors to be used by the flow solver, a positive non-unit power of 2, is selected. PREMG usually takes a few minutes and runs on only one CPU. Message passing is organized around edges that cross boundaries between subdomains. PREMG creates a large file, named *premg.unf*, containing the mesh information for the flow solver.

The iterative part of the AIRPLANE flow solver, named MIDMPIMG, applies the Runge-Kutta scheme to find a steady Euler flow solution. It uses the MPI message passing library and multigridging. The optimization program passes run control and convergence parameters to the flow solver through a text file. The surface pressures and flow field solution may be stored in files at the end of the run. MIDMPIMG usually takes much more wall time than APREPHIS, AP-CATMSH, and PREMG. Post-processing utilities may optionally be applied in order to carefully examine the flow field.

#### **AIRPLANE MULTIGRID ACCURACY AND SPEEDUP**

Performance of AIRPLANE when using 3-mesh multigridging with a W cycle is examined for a transport aircraft and for a baseline CTV across a range of Mach numbers. Accuracy and speedup for several multigridging cases across subsonic to transonic Mach numbers for the transport, and subsonic to high supersonic for CTV, are described in this section.

## TRANSPORT AIRCRAFT

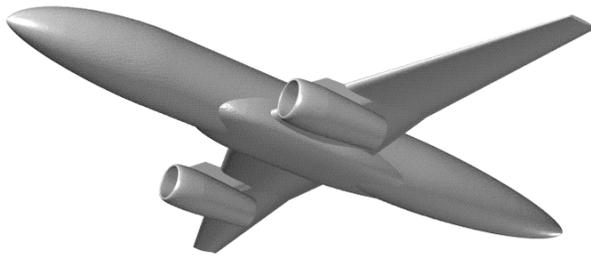


Fig. 9. Transport Aircraft, Mesh 6.

The transport is a simple wing-body with an underwing pylon and flow-through nacelle (Fig. 9). All cases were run at a constant angle of attack of  $-1.0^\circ$  and  $0.0^\circ$  yaw. Unstructured meshes of 7 different sizes were prepared. The number of points in the first mesh is 29,184. The sixth mesh has 1,955,784 points. Relative mesh sizes are shown in the following bar chart, Fig. 10.

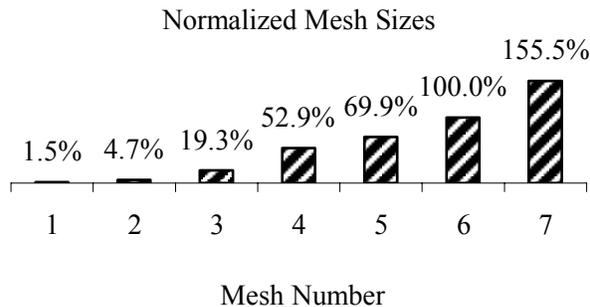


Fig. 10. Transport Mesh Sizes Relative to Mesh 6.

Solutions were generated for increasing Mach numbers up to 0.78, or until the flow solver returned unsteady or unstable results. Convergence parameters were set at the beginning and left alone; it might be possible to extend some of the cases to higher transonic Mach numbers by changing convergence parameters.

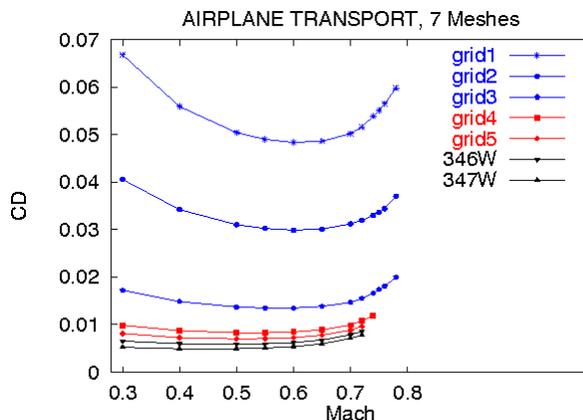


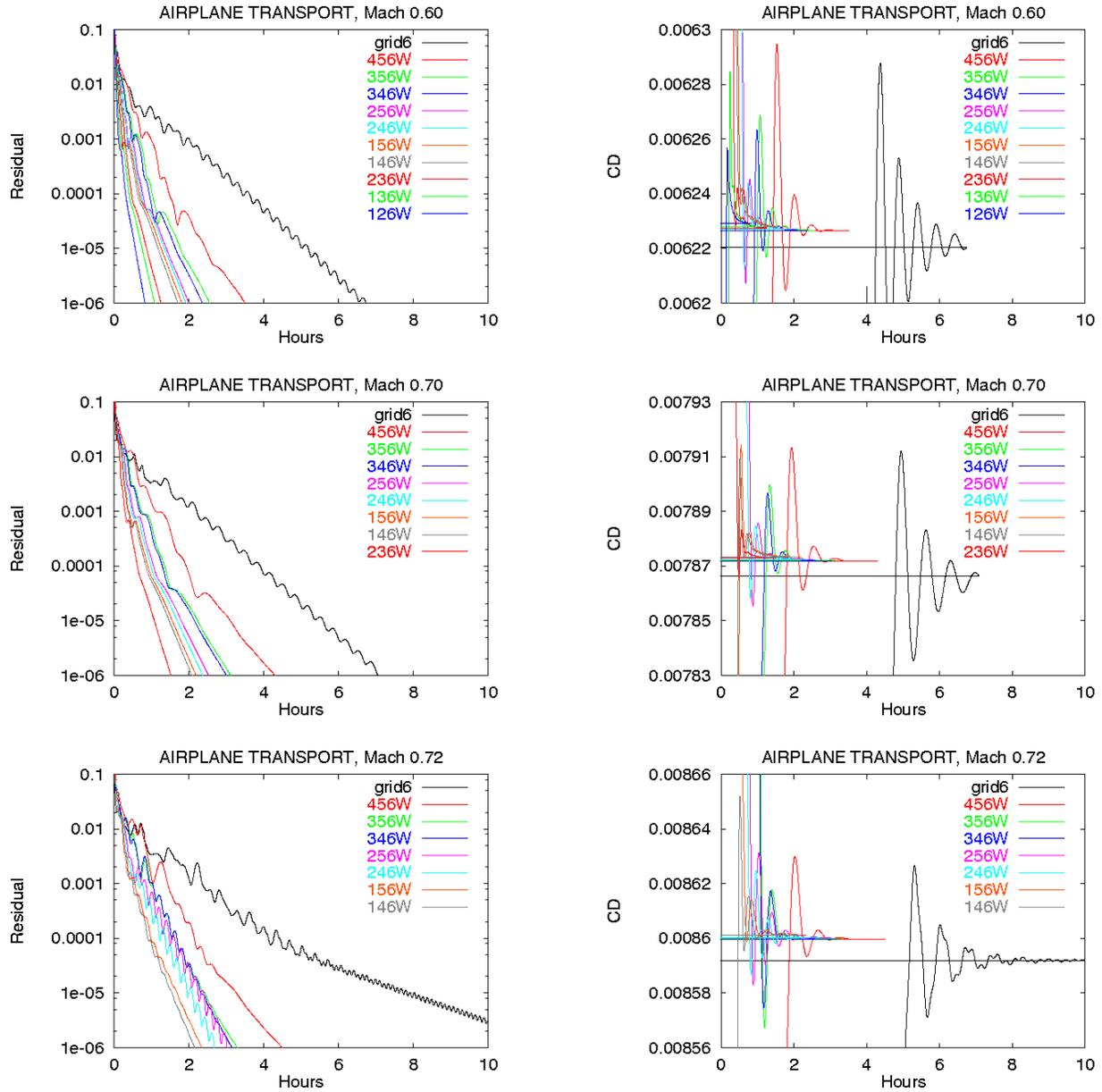
Fig. 11. AIRPLANE Transport Drag Predictions.

Drag predictions for all 7 meshes across the range of successful Mach numbers are shown in Fig. 11. Curves are ordered top to bottom for meshes 1 to 7. The curves get closer together as the meshes get finer. The finer meshes encounter unsteady flow behavior near the pylon trailing edge at lower transonic Mach numbers. The first three transport meshes produce inaccurate solutions when used alone, but can be useful when multigridding. The predictions in the figure use no multigridding for the first five meshes, then multigridding for the last two (346W and 347W). The curve name indicates which meshes and cycle type were used. For example, 346W indicates a solution on mesh number 6 using meshes 3, 4, and 6, in a W cycle.

Accuracy of the final prediction depends upon the finest mesh as well as the collection of meshes used. The latter issue was studied in detail, using all combinations of three meshes ending with mesh 6.

Fig. 12 shows how convergence of residual and drag vary across the ten possible multigridding cases, plus mesh 6 without multigridding, at three different Mach numbers, 0.6, 0.7, and 0.72, for the transport. The curves are ordered top to bottom by decreasing flow solver wall time required to achieve an average residual of  $10^{-6}$  in each figure. The horizontal axis is hours, scaled 0 to 10 in each plot. Every case uses 32 CPUs, each at 400 MHz, on an SGI Origin 2000 at NASA Ames. Total mesh size, wall time, drag, and lift are tabulated at the bottom of the figure for every case.

The force coefficients obtained with or without multigridding indicate that the solution accuracy is not affected via the multigridding approach. Lift predictions differ by 0.0001, and drag predictions differ by about one tenth of a count, for all three Mach number cases. Cases marked "Unst." are either unsteady or unstable for the set of meshes and/or convergence parameters used.



Meshes & Cycle	Total Points	Mach 0.60			Mach 0.70			Mach 0.72		
		hours	CD cts	CL	hours	CD cts	CL	hours	CD cts	CL
Mesh 6	1955784	6.754	62.205	.220365	7.092	78.665	.245895	11.856	85.920	.255108
456 W	4356896	3.509	62.263	.220349	4.305	78.717	.245843	4.506	85.995	.255046
356 W	3701263	2.562	62.265	.220342	3.123	78.719	.245834	3.303	85.998	.255037
346 W	3367821	2.368	62.265	.220343	3.010	78.718	.245836	3.163	85.997	.255039
256 W	3415536	2.007	62.270	.220332	2.529	78.724	.245821	3.027	86.003	.255023
246 W	3082094	1.929	62.270	.220333	2.383	78.723	.245822	2.697	86.002	.255025
156 W	3352245	1.816	62.279	.220319	2.202	78.734	.245801	2.349	86.013	.255001
146 W	3018803	1.712	62.279	.220320	2.085	78.733	.245802	2.163	86.012	.255002
236 W	2426461	1.261	62.275	.220318	1.526	78.729	.245804	Unst.	N/A	N/A
136 W	2363170	1.092	62.279	.220311	Unst.	N/A	N/A	Unst.	N/A	N/A
126 W	2077443	0.831	62.292	.220285	Unst.	N/A	N/A	Unst.	N/A	N/A

Fig. 12. AIRPLANE Transport Multigrid Convergence at Three Mach Numbers

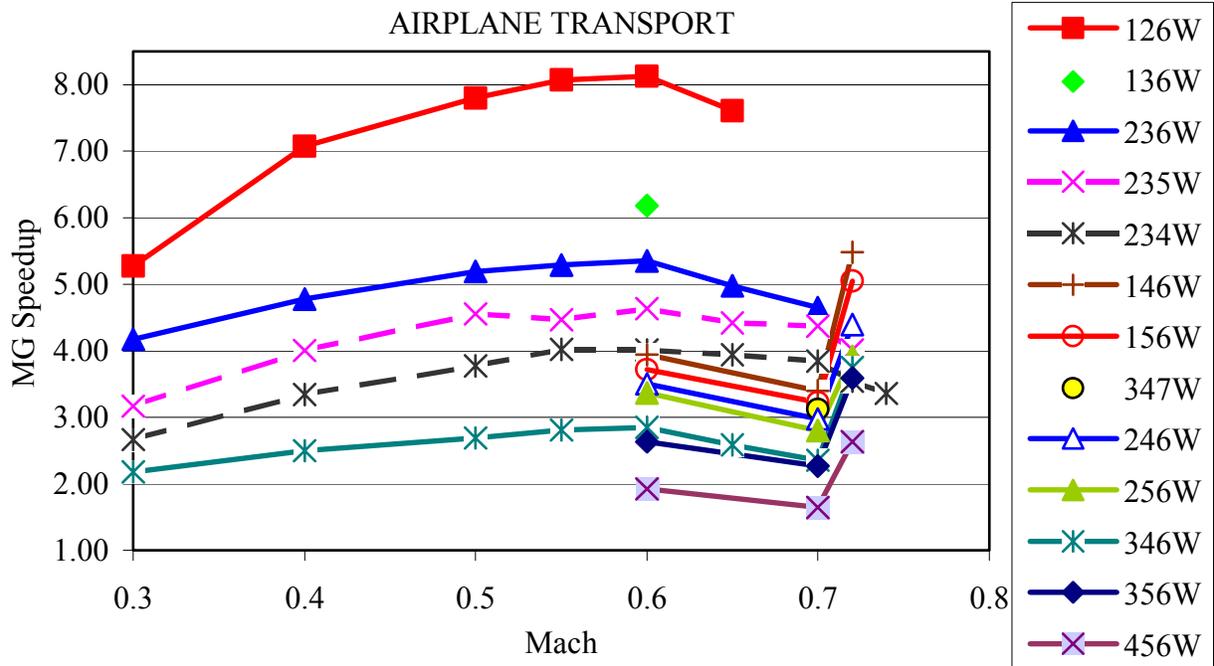


Fig. 13. Multigrid Speedup for the Transport Aircraft.

The largest multigridding speedup, relative to the single grid solution on mesh 6, was obtained when the coarse meshes were much coarser than mesh 6. The 126W case was over 8 times faster at Mach 0.6, for example, but it did not work at Mach numbers above 0.65. Multigrid speedup for the transport is plotted in Fig. 13, for Mach numbers ranging from 0.3 to 0.74. The curve labels are ordered top to bottom to match the order of the curves, with regard to the speedup ratio.

Each point in the plot required two runs of the flow solver, one with and one without multigridding. The ten experiments that ran at Mach numbers 0.6, 0.7, and 0.72 for the three combinations ending with mesh number 6 are plotted with solid lines. Speedups for these cases are ratios of wall times (hours spent in the iterative part of the flow solver) seen in the table of Fig. 12. Multigrid speedup increases sharply between Mach 0.7 and 0.72 in many cases, probably because of extra numerical dissipation which is present when multigridding. The solver without multigridding has a harder time converging at Mach 0.72 because of a strong shock, which is present at the trailing edge of the pylon.

Results of three other measurements for meshes 4, 5 (dashed lines), and 7 (circle, one point) are presented in the same plot for comparison. Case 235W goes from Mach 0.3 to 0.72, case 234W goes from Mach 0.3 to 0.74, and case 347W has only one point at Mach 0.7 with a speedup of 3. The result for mesh number 7, which has over 3 million points, took 14.2 hours with-

out multigridding, but only 4.6 hours with multigridding.

Five of the speedup curves start at Mach 0.3 and go to Mach 0.7, 0.72, or 0.74, after which the results were unsteady or unstable.

Using finer meshes on the coarse end tends to be more robust, but it is much slower. The 236W case ran to Mach 0.7, but it is only 4-5 times faster.

**BASELINE CTV**

Multigridding results for the CTV baseline configuration were also examined for accuracy and speedup, but across a wider range of 11 Mach numbers, from 0.3 to 8, at alpha 5°.

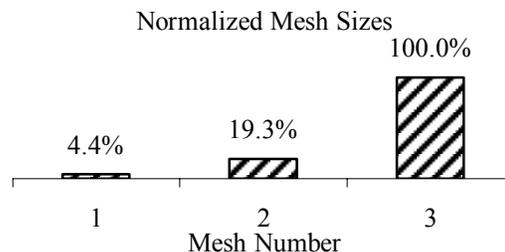


Fig. 14. CTV Mesh Sizes Relative to Mesh 3.

The single grid case used a fine mesh of 324,475 points. Only one multigrid case was examined, using coarse and medium meshes of 14,234 and 62,687 points, respectively, and the same fine mesh as before. Mesh

sizes are shown in Fig. 14, scaled relative to the fine mesh.

Convergence parameters were tuned for each case, and each run stopped at residual  $5 \times 10^{-5}$  (approximately 5 orders of magnitude reduction of the average residual). Suitable multigridding parameters were not found beyond Mach 4 (solution divergence or no computational time improvement when multigridding). Drag predictions are plotted in Fig. 15. The use of multigridding has little effect on the solutions. The drag prediction changed by an average of 2 counts. The largest difference was 7 counts at Mach 2 (1%). At Mach 0.8 the difference was 0.4 drag counts.

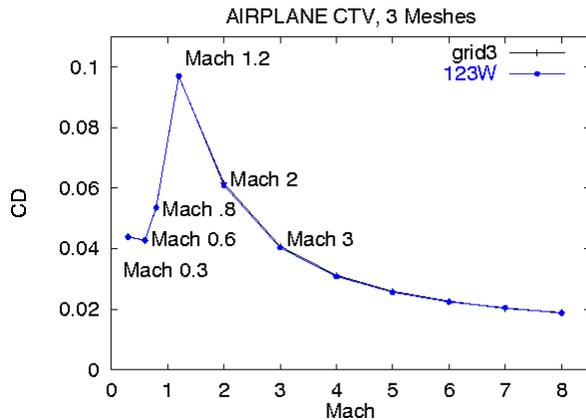


Fig. 15. AIRPLANE CTV Drag Predictions.

When multigridding worked, it achieved a speedup of at least 2, with a speedup of 12 at Mach 0.8, see Fig. 16.

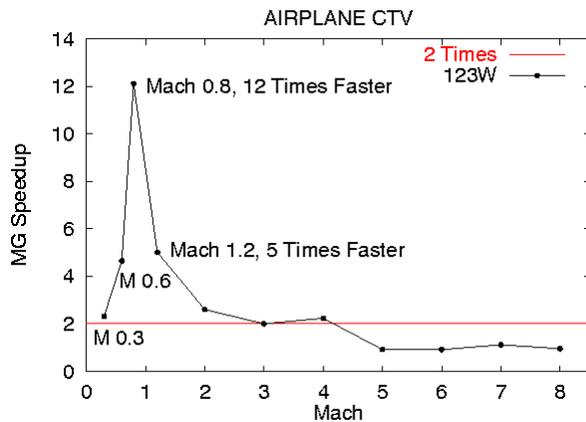


Fig. 16. Multigrid Speedup for Baseline CTV.

#### OPTIMIZATION COMPUTATIONAL TIME ESTIMATES

An example of the wall and computational time will be reported for a three design point optimization for the CTV. The goal was to improve the performance and handling characteristics compared to a baseline configuration, by applying small to moderate shape changes to the wing and body. The three Design Points (DP), are shown in Table 1, below.

A full span solution was required for DP 3, with a  $10^\circ$  side slip angle. Multigridding (M.Grid) was successfully used for the Mach 0.3 cases, but was not successful at Mach 6. The Mach 6.0 solution was obtained solely on the fine mesh. Approximate flow solver wall times (seconds) when using 64 CPUs on a 300 MHz SGI Origin 2000 are shown in the rightmost column.

DP	M	$\alpha$	$\beta$	Iter.	Span	M.Grid	Sec.
1	0.3	8.5	0	200	Half	Yes	381
2	6.0	13.1	0	1000	Half	No	470
3	0.3	10.0	10	200	Full	Yes	580

Table 1. CTV Design Points

Note that the full span solutions required less than double the half span solution times. Parallelization is more efficient for the full span case because of the reduction in the ratio of inter-domain communication time compared with domain processing time. The flow solver generates these three solutions in about 24 minutes. A single-point design requiring half-span computations would likely take one-quarter of this time, or about 6 minutes.

Seven design variables were used. These include wing twist, fore/aft body droop, forebody camber and forebody upper surface thickness. The objective function is a sum of six terms, shown below. The weights  $\sigma_1$ - $\sigma_6$  were chosen based on the starting solutions at each DP(1-3) and the importance of each term. Typically  $\sigma_n$  is chosen such that each term is equal to 1.0 and then scaled based on importance of each term relative to the others as determined by the designer.

Objective function =

$$\sigma_1 C_D(DP1)/C_L(DP1) + \sigma_2 C_D(DP2)/C_L(DP2) + \sigma_3 (C_M(DP1) - 0.0001)^2 + \sigma_4 (C_M(DP2) - 0.0001)^2 + \sigma_5 (C_l(DP3) + 0.01)^2 + \sigma_6 (C_n(DP3) - 0.01)^2$$

The first two terms address aerodynamic performance, the next two are pitching moment penalty terms to address vehicle trim, and the last two are rolling and yawing moment penalty terms.

The optimization process required 127 function evaluations to complete 10 design iterations. Each evaluation involved perturbing the mesh and computing three flow solutions, one for each design point.

The time required by the optimization program QNMDIF is insignificant. Wall clock time is dominated by function evaluation, that is, the time it takes to analyze a given shape at the three design points for this case. Flow solutions for each design point are generated sequentially. AIRPLANE's domain decomposition programs recursively splits the computational mesh in half requiring the number of processors (64 in this case) to be a power of 2. The main steps and approxi-

mate wall times for this multipoint optimization are shown below:

<u>Runs</u>	<u>Operation</u>	<u>CPUs</u>	<u>Time</u>
3	Perturb surface mesh	1	26 s.
3	Move volume mesh	16	125 s.
8	Find interp. coeff.s	16	207 s.
2	Mesh preprocessing	1	543 s.
3	Run the flow solver	64	1431 s.
	Total		2332 s.

Table 2. Operations and Wall Clock Times

Surface mesh perturbations include a thickness check, and are run sequentially for the three surface meshes. This takes a total of about 26 seconds.

Volume mesh perturbation is parallelized and is usually much faster than regenerating the mesh from scratch. Three meshes are moved at the same time. Wall time is dominated by the finest mesh, the other two perturbations are essentially free. In one case it took about 125 wall seconds, 1506 seconds CPU time on 16 CPUs (12× speedup) for the finest mesh; the other two meshes finished earlier.

Computation of the interpolation coefficients is parallelized, four sets of coefficients are required for three level multigridding, and each set may be computed simultaneously. Typically four 16-CPU processes run at the same time, the interpolations are rapid between the coarse meshes and with substantially greater CPU time requirements for interpolations involving the finest mesh. It takes about 69 wall seconds on the finest mesh, 676 CPU seconds on 16 CPUs (10× speedup). These numbers are for a half-span mesh, and the results are used for the first two design points. The full-vehicle mesh is obtained by mirroring the half-span mesh, but interpolation coefficients are recomputed.

The flow solver requires a scalar preprocessing step to set up recursive bisection domain decomposition used by MPI. It takes about 181 seconds to preprocess a non-yawed mesh. This step only needs to be done once for the first two (half span) design points. It takes about twice as long for the full span mesh.

The three design points have different computational requirements (Table 1), about 1431 seconds total. The flow solver uses 3-mesh multigridding for design points 1 and 3, and no multigridding (computation on just the finest mesh) for design point 2. The mesh and computational requirements are less than doubled for design point 3 in order to compute rolling and yawing moment coefficients with a full span mesh at non-zero sideslip.

The optimization was stopped after 127 function evaluations, during the eleventh optimization iteration when improvements were no longer achieved. Assuming the mesh movement procedure always had been successful, the wall time would have been about 83 hours, or three and a half days. (It takes about 39 minutes wall time per function evaluation, Table 2). A more conservative estimate would be four days to repeat this case.

A single-point optimization such as design point 1 (Mach 0.3) would require approximately 21 minutes per function evaluation, or 45 hours (1.9 days) for the 127 function evaluations used for the aforementioned optimization problem.

### OPTIMIZATION EXAMPLE

An example of the AIRPLANE optimization process will be shown for a design modification of a CTV configuration in which wind tunnel data, Navier-Stokes computations, and flight simulation data were obtained for the baseline and optimized configurations. AIRPLANE was chosen since it was found to be more robust than FLOWCART. Previous use of FLOWCART and AIRPLANE optimization on a proprietary vehicle found that FLOWCART would become unstable during the course of optimization. The CFL number input was periodically modified manually to obtain convergence and optimization was restarted. Very small changes to the CFL number (values between 1.3 and 1.5) were found to correct the convergence problem. Hence, FLOWCART would require an automated method to change the CFL number when the optimization method detected no convergence. AIRPLANE, in contrast, did not exhibit any convergence failures during optimization.

The AIRPLANE-based optimization example is somewhat academic in nature since the wing twist and camber were modified without appropriate structural or weight constraints. It is often quite informative to optimize without constraints to determine the design variable sensitivity, and to determine constraint locations, before starting the design in earnest.

The CTV was optimized at two design points (Mach 0.3,  $\alpha=9^\circ$ , and Mach 6.0,  $\alpha=13^\circ$ ). The design objective was to modify the longitudinal stability so that the vehicle would be trimmed or nearly trimmed while simultaneously improving the performance of the vehicle at both design points. The objective function was of the form:

$$\text{Objective Function} = \sigma_1 (C_D/C_L) + \sigma_2 (C_M - 0.0001)^2$$

where  $\sigma_1=10$ , and  $\sigma_2=10,000$ . The wing twist, camber and elevator (wing flap) deflection angle were modified. This required 13 wing design variables, 1 twist, 10 camber variables, (5 sine functions distributed at the

break and 5 along the tip of the wing), and 2 elevator (wing flap) deflection angles. The wing flap angle was determined independently of the camber and twist design variables at the two Mach numbers, thus different geometries were analyzed at each design point during optimization, since separate meshes were required for the different flap settings at the two Mach numbers. The wing tip camber modifications were lofted inward to the wing break, and the break section modifications were lofted to the tip and inward to near the side of the body. It is possible to carry the wing modification inboard of the body intersection, however, when wing and body components are altered independently, poor quality triangles are formed near the intersection, making the surface mesh unsuitable for AIRPLANE. On the other hand, FLOWCART can handle this situation.

The optimized configuration showed significant camber changes combined with  $12.4^\circ$  of twist (wash out) about the wing trailing edge. The camber at the break was deemed excessive, since the upper surface was concave. The upper surface of the break section was modified manually to eliminate the concavity. The resulting design is shown in Fig. 17 (compare with Fig. 1). The optimized elevator deflection angles were  $0.05^\circ$  (flap down) and  $-5.2^\circ$  (flap up) for the Mach 0.3 and Mach 6.0 design points, respectively. Plastic models of this configuration and the baseline configuration were built for evaluation in the Ames 32 inch by 48 inch atmospheric low speed wind tunnel (Mach 0.11).<sup>24</sup> The scale factor was 1/20 of the full size configuration, resulting in a model length of 22 inches.

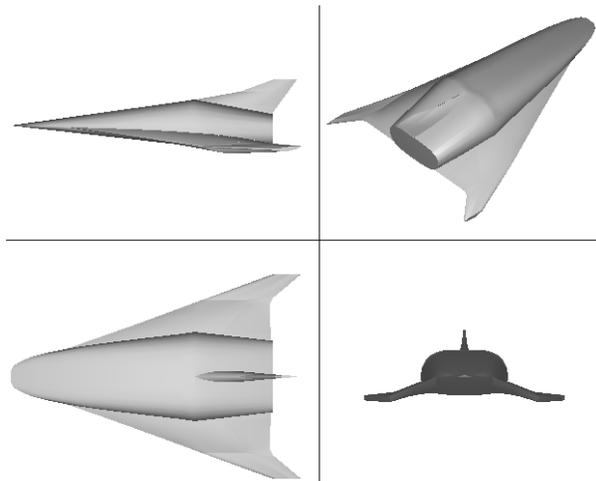


Fig. 17. Optimized CTV Configuration.

The moment coefficients for the baseline and optimized configurations are shown in Figs. 18 and 19, respectively. Computations from FLOWCART, OVERFLOW,<sup>25,26</sup> and the experimental data can be compared.

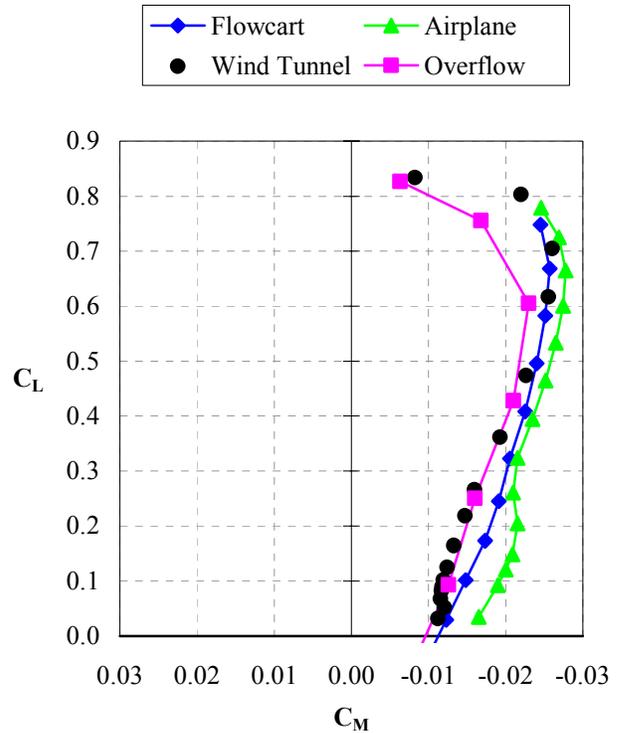


Fig. 18. Baseline CTV Configuration, Mach  $\leq 0.3$ .

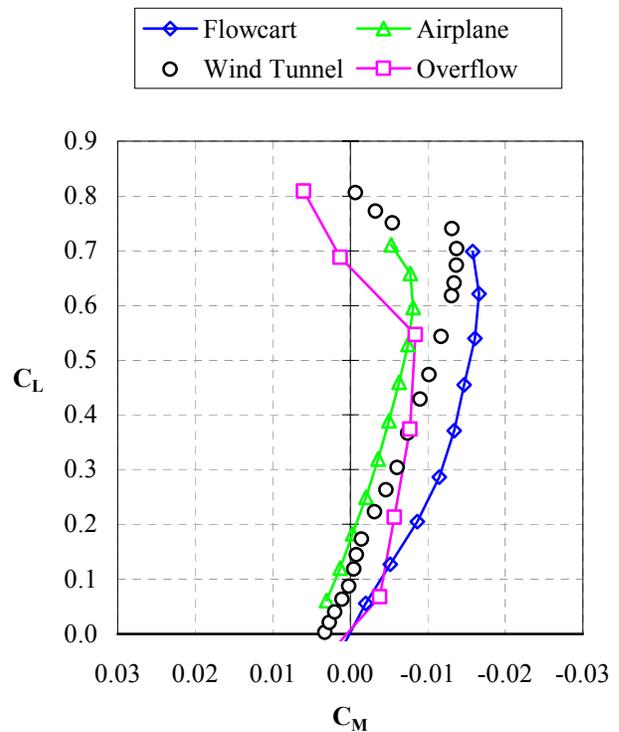


Fig. 19. Optimized CTV Configuration, Mach  $\leq 0.3$ .

The optimized vehicle has greatly improved longitudinal trim characteristics compared to the baseline configuration (compare Figs. 18 and 19). Although the optimization objective fought to reduce drag, the drag coefficient increased approximately 190 counts or 16% (not shown).

The CFD computations were obtained at Mach 0.3, without the flap deflections obtained with optimization. The experimental data were taken at Mach 0.11 (also without flap deflections), with a Reynolds number of 1.1 million based on body length. The subsonic Mach number difference between the computational and experimental data should have minimal effects on the force and moment coefficient data.

The Navier-Stokes viscous computations were at the wind tunnel Reynolds number (0.5 million per foot) using the k- $\epsilon$  turbulence model.

The wind tunnel data do not include a base drag or cavity correction. The Euler and Navier-Stokes computational data include the base in the integration of force and moment coefficients, without any special treatment of the base.

Hypersonic characteristics for the second design point at Mach 6 were computed using AIRPLANE, FLOWCART and OVERFLOW. Moment coefficient computations of the baseline and optimized configurations are shown in Fig. 20. Here, the OVERFLOW computations were near flight conditions at a Reynolds number of 2 million per foot and the Baldwin-Barth turbulence model was used.

Note that the predictions are not plotted to the same scale as the Mach 0.3 results. The aerodynamic changes resulting from optimization are small at Mach 6.0 in comparison with the subsonic results. This is due to the nearly constant upper surface pressures at high Mach and angle of attack. However, the trim characteristics of the optimized configuration show improvement for both the Euler and Navier-Stokes computations compared with the baseline. The configurations would be closer to trim if the computations had included the  $-5.2^\circ$  flap deflection determined during optimization. The changes to the drag coefficient were minimal, with a 15 count or 5% increase over the baseline (not shown).

#### **PILOTED SIMULATIONS**

Navier-Stokes and Euler computational analyses were used as data for the Ames Vertical Motion Simulator. NASA astronaut piloted time-history flight data for the baseline and optimized configurations are shown in Fig. 21 for approach and landing flight starting from an altitude of 5000 feet at a velocity of 250 knots. Spoilers (a split rudder) were necessary to reduce the velocity of

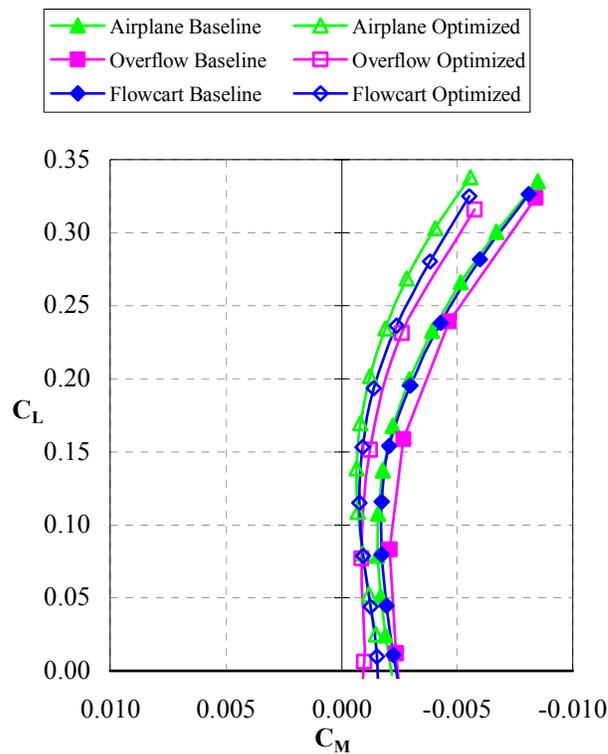


Fig. 20. Baseline and Optimized Configurations, Mach 6.0.

the vehicles. However, upon deflection they increase the pitching moment (nose up) of the vehicles compared to the computations in Figs. 18 and 19. The improvement in trim characteristics of the optimized configuration is shown by the reduction of the magnitudes of the elevator deflections used, compared to the baseline configuration when the spoilers are not deflected (after 50 seconds).

The optimization example just reported is one of four AIRPLANE-based designs of the CTV configuration that were assessed in the VMS using Cooper-Harper ratings<sup>27</sup> for the handling quality assessment. AIRPLANE was further used to re-design one of the four vehicles during a two-week period. Five separate optimizations from which a single design was selected were performed during this period.<sup>28</sup> This rapid redesign allowed the astronauts to assess a 5<sup>th</sup> configuration during the one-month entry in the flight simulator. Four of the five optimized configurations obtained Cooper-Harper ratings of 2 (good/negligible deficiencies), and one of them was rated at level 1 (excellent/highly desirable).

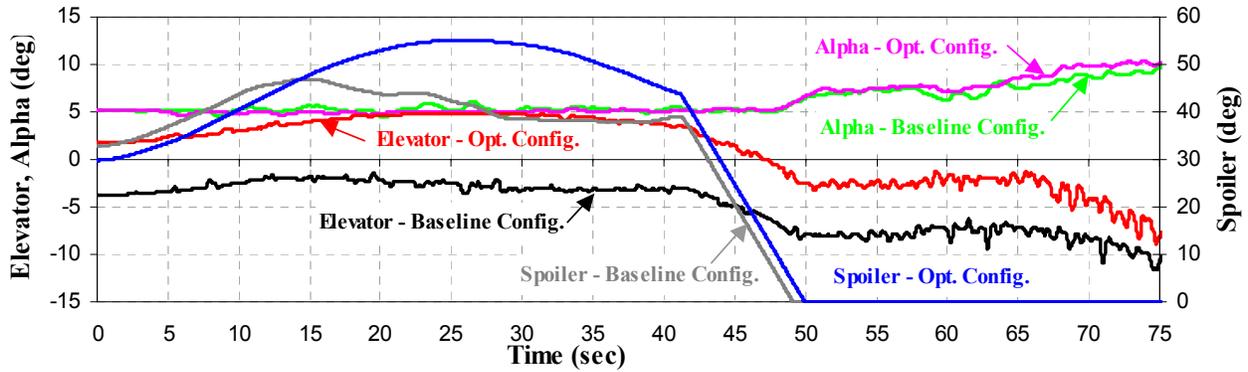


Fig. 21. VMS Flight Simulation Data for Approach and Landing from 5000 feet at 250 knots.

### AIRPLANE/FLOWCART GRADIENT AND SOLUTION COMPARISONS

Prior to the design of the CTV, FLOWCART and AIRPLANE were used to study the X-37 configuration. These optimization tools were used in an attempt to improve the stability and control and lift/drag ratio of the configuration. The design variables in this example are the multipliers of sine-based shape functions applied to the wing surfaces. Forward and backward finite difference calculations for each design variable and for each code are shown in Fig. 22.

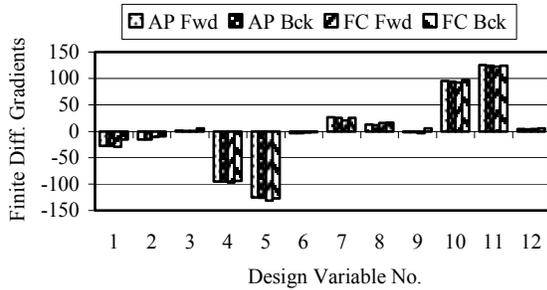


Fig. 22. RLV Finite Difference Gradients.

Confidence in finite difference gradient accuracy is high when the forward and backward gradients have the same sign and have approximately the same magnitude. Notable differences in the gradient values between AIRPLANE and FLOWCART led to an examination of the second derivatives calculated based upon the forward and backward first derivatives (see Fig. 23).

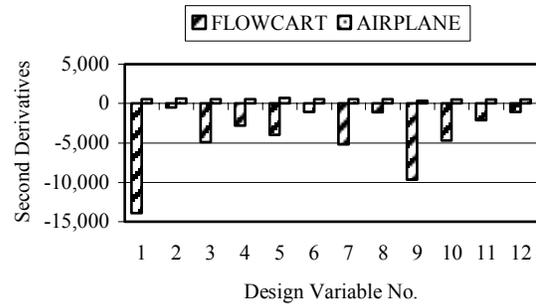


Fig. 23. RLV Second Derivatives.

The negative values of the second derivatives seen in the FLOWCART solutions pose a significant problem from an optimization standpoint since they indicate a convex (and possibly multi-modal) design space. While the ill-posed condition is corrected by QNMDIF through a simple change of sign, the convexity often implies a more fundamental problem. Negative second derivatives can occur for the following two reasons. First, that the design problem is poorly formulated (i.e. that it is truly multi-modal). Second, the negative values can arise from inaccurate calculations of the finite difference gradients themselves. In the problem at hand, the objective function is a combination of minimizing the drag-to-lift ratio and trim. Both of these functions are usually concave with respect to vehicle shape and hence it is highly improbable that second derivatives with respect to several design variables would all have negative signs. A more likely cause for the negative second derivatives is in the finite difference calculations for the FLOWCART cases. There are two possibilities. First, the fact that there is no easy way to use a “smooth” mesh movement strategy when calculating gradients for a Cartesian mesh can lead to problems with volume mesh points appearing and disappearing during re-meshing for the gradients. Second, the level of convergence of the FLOWCART solutions may not have been sufficient for either the initial solu-

tion or for the subsequent perturbed solution to render accurate gradients. More detailed studies will need to be performed to determine the actual cause.

In spite of the differences in derivative computations with AIRPLANE and FLOWCART, QNMDIF proceeds without difficulty for both codes. The optimizations after one complete design iteration with the two codes yielded very similar results. The values of the design variables and the percentage reduction (gain) of the objective function after one design iteration are compared in Fig. 24. The fact that both methods were able to make similar refinements after one iteration implies a success when there was plenty of room for improvement. A better test for an optimization method occurs when improvements are very difficult to obtain and only very accurate gradients can reveal the correct direction to proceed.

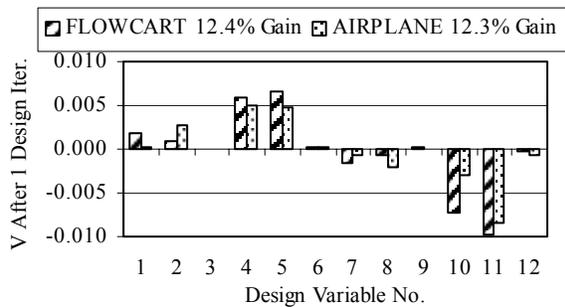


Fig. 24. RLV Design Variables.

For further evaluation of FLOWCART-based optimization, a simplified four-design-variable single-point (Mach 1.2,  $\alpha=10^\circ$ ) design problem was set up for the CTV configuration. The design variables were wing twist (variable 1) from the root to tip and the camber at the wing break lofted to the tip and the side of body. The camber modification used three design variables located at the 30, 50, and 70 percent chord locations (variables 2, 3, and 4). The identical design problem was run with AIRPLANE for comparison purposes. The gradient and second derivative computations for the two codes are shown in Figs. 25 and 26. The forward and backward gradients computed with AIRPLANE are more consistent than those computed via FLOWCART. However, FLOWCART's second derivative computations have improved compared with the computations for the RLV design, with 3 of the 4 second derivatives now positive. This improvement may be due to a finer mesh with increased refinement near the surfaces of the configuration, that was generated using a new feature in the Cartesian mesh generator that refines near the surface, based on the size of the surface triangulation used for AIRPLANE. The results after three design iterations are shown in Fig. 27. FLOWCART and AIRPLANE show similar design

variable values and comparable design improvements. Hence, this problem offers a second and better founded demonstration that FLOWCART is quite capable of gradient-based numerical optimization.

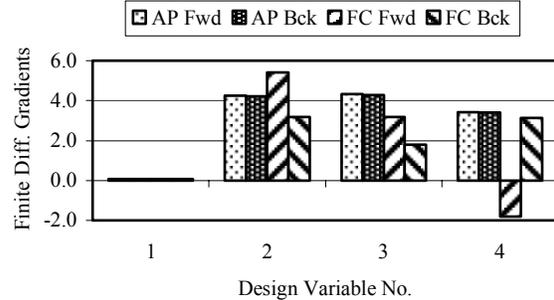


Fig. 25. CTV Finite Difference Gradients.

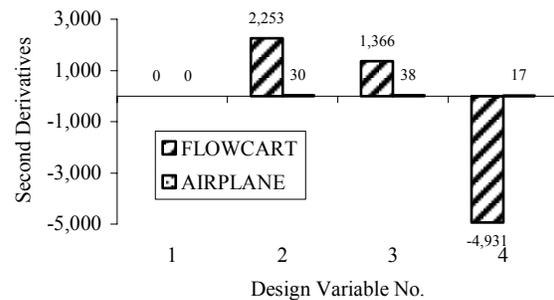


Fig. 26. CTV Second Derivatives.

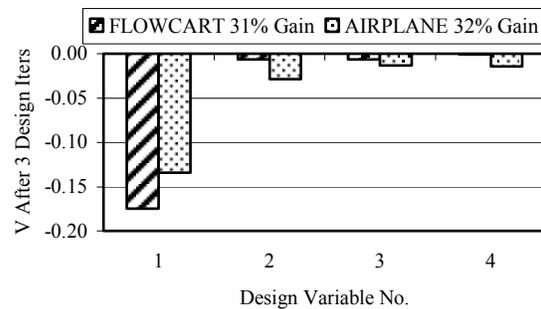


Fig. 27. CTV Design Variables.

The two comparisons shown were for aerospace vehicles with bluff fuselage bases that produce unsteady separated flow. The application of an Euler code to such a viscous-dominated problem is questionable, and convergence is difficult to achieve. However, it is evident from the computations that the numerical dissipation used to stabilize the convergence for Euler computations can simulate, to a limited extent, the effects of viscosity. It is recognized that this is non-physical, but Navier-Stokes based optimization is much too costly.

AIRPLANE was able to achieve convergence of approximately 5 orders reduction when multigridding was employed. While the upwind scheme in FLOWCART produces crisp shock resolution, the minmod limiter used with it fluctuates in near-constant portions of the flow field preventing deep convergence of the residual norms.<sup>29</sup> However, FLOWCART's force and moment coefficients do rapidly stabilize. These facts considered alone would still yield a questionable result, but the agreement with the converged AIRPLANE solutions helps to affirm the FLOWCART solutions. The lack of residual convergence with FLOWCART and the use of different grids with each new design variable are thought to be the primary reasons for the slightly poorer gradient computations compared with AIRPLANE. If the flow phenomena were without strong shocks or separation, FLOWCART would likely produce improved gradients.

#### **CONCLUDING REMARKS**

Two optimization tools were developed using AIRPLANE and FLOWCART, and both were successful in making design improvements. These tools were employed on aerospace vehicles with difficult flow physics, but they are also well suited for any type of aircraft.

AIRPLANE proved to be extremely robust and did not fail during the optimizations shown in this report and several other designs not reported herein. FLOWCART, however, required periodic modification of the input variables in order to proceed. Although one can automate the hand manipulations of the input file, a robust code will probably prove to be the more efficient approach.

Tools were developed to make these optimization techniques computationally efficient. This includes the development of a method to perturb the surfaces of wing and body type components (used by AIRPLANE and FLOWCART) and volume mesh movement schemes (used by AIRPLANE). The mesh movement techniques were fairly successful. They were effective approximately 85% of the time. The gradients were better when the meshes were deformed rather than regenerated, and, when successful, the methods reduced the grid generation wall time by 71 percent when 16 CPUs were used.

A multigrid capability was added to the flow solver within AIRPLANE and was evaluated for a range of Mach numbers on a full component transport and CTV configuration. The multigrid option significantly reduced the computational requirements to achieve convergence. Typical speedup factors of 3 to 5 were observed.

The multigrid studies suggest that the size of the coarser grids should be reduced by factors of 8 to 20

between successive grids for increased speed up, but coarsening the meshes too rapidly can compromise the robustness of the code.

#### **OBSERVATIONS / RECOMMENDATIONS FOR METHOD IMPROVEMENTS**

Central difference gradient computations are expensive. An adjoint method could be employed for these computations in the future to reduce the number of computations to the equivalent of two flow solutions for any number of design variables (provided it is parallelized and uses multigridding).

Constraints implemented through the use of penalty terms added to the objective function create rapid changes in the design space and cause the method to take many search steps to find a direction for an improvement. QNMDIF should be replaced by a constrained gradient method such as NPSOL<sup>30</sup> in the future.

The FLOWCART-based optimization method allows for the component intersections to be changed during the course of optimization. The AIRPLANE-based method, however, needs an enhanced mesh movement or deformation method to improve the surface triangulation quality in the vicinity of new intersections, with properly formed tetrahedra near the intersected surfaces automatically generated.

#### **ACKNOWLEDGEMENTS**

The authors would like to thank Michael Aftosmis, Fanny Zuniga, John Melton, David Kinney, James Reuther and Stephen Smith (NASA), Chun Tang, Veronica Hawke and David Saunders (Eloret) for their guidance and contributions to this report.

#### **REFERENCES**

1. Cliff, S. E., Reuther, J. J., Saunders, D. A., and Hicks, R.M., "Single-Point and Multipoint Aerodynamic Shape Optimization of High-Speed Civil Transport," AIAA Journal of Aircraft Volume 38, Number 6, Pages 997-1005, November-December 2001.
2. Jameson, A., "Optimum Aerodynamic Design via Boundary Control," RIACS Technical Report 94.17, Princeton University Report MAE 1996, Proceedings of AGARD FDP/Von Karman Institute Special Course on "Optimum Design Methods in Aerodynamics," Brussels, April 1994, pp. 3.1-3.33.
3. Jameson, A., and Martinelli, L., "Aerodynamic Shape Optimization Techniques Based on Control Theory," CIME (International Mathematical Summer Center), Martina Franca, Italy, June 1999.
4. Reuther J., Jameson, A., Farmer J., Martinelli L., and Saunders D., "Aerodynamic Shape Optimization of Complex Aircraft Configurations via an Adjoint Formulation," AIAA Paper 96-0094, January 1996.

5. Reuther, J., Alonso, J., Rimlinger, M., and Jameson, A., "Aerodynamic Shape Optimization of Supersonic Aircraft Configurations via an Adjoint Formulation on Distributed Memory Parallel Computers," AIAA Paper 96-4045, September 1996.
6. Kinney, D.J., Bowles, J.V., Yang, L.H., and Roberts, C.D., "Conceptual Design of a 'SHARP'-CTV," AIAA Paper 2001-2887, June 2001.
7. Kolodziej, P., Bull, J., Salute, J., Yang, L., and Roberts, C., "First Flight Demonstration of a Sharp Ultra-High Temperature Ceramic Nosetip, NASA TM 112215, December 1997.
8. Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, 4th edition, Academic Press, Inc., San Diego, 1982.
9. Gill, P.E., Murray, W., and Pitfield, R.A., "The Implementation of Two Revised Quasi-Newton Algorithms for Unconstrained Optimization," NAC-11, April 1972.
10. Jameson, A., "Aerodynamic Design Via Control Theory," *Journal of Scientific Computing*, 1988, pp. 233-260.
11. Jameson, A., "Automatic Design of Transonic Airfoils to Reduce the Shock Induced Pressure Drag," *Proceedings of the 31<sup>st</sup> Israel Annual Conference on Aviation and Aeronautics*, Tel Aviv, Feb. 1990, pp. 5-17.
12. Lions, L. J., *Optimal Control of Systems Governed by Partial Differential Equations*, Springer-Verlag, New York, 1971. Translated by S.K. Mitter.
13. Pironneau, O., *Optimal Shape Design for Elliptical Systems*, Springer-Verlag, New York, 1984.
14. Shubin, G. R., "Obtaining Cheap Optimization Gradients from Computational Aerodynamics Codes," International paper AMS-TR-164, Boeing Computer Services, June 1991.
15. Jameson, A., and Baker, T.J., "Improvements to the Aircraft Euler Method," AIAA Paper 87-0452, January 1987.
16. Jameson, A., Cliff, S., Thomas, S., Baker, T., and Cheng, W., "Supersonic Transport Design on the IBM Parallel System SP-2," *Computational Aerosciences Workshop 95 Proceedings*, March 7-9, 1995, pp. 78-81.
17. Baker, T. J., and Vassberg, J. C., "Tetrahedral Mesh Generation and Optimization," *Proceedings of the 6<sup>th</sup> International Conference on Numerical Grid Generation*, Greenwich, UK, July 1998, pp. 337-349.
18. Aftosmis, M., "A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries," AIAA paper 2000-0808, January 2000.
19. Aftosmis, M., "Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry," AIAA paper 97-0196, January 1997.
20. Baker, T., "Mesh Modification for Solution Adaptation and Time Evolving Domains," Proceedings of 7<sup>th</sup> International Conference on Numerical Grid Generation, Chateau Whister BC, September 2000.
21. Baker, T., "Mesh Movement and Metamorphosis," Proceedings of 10<sup>th</sup> International Meshing Round Table, Newport Beach, CA, October 2001.
22. Jameson, A., "Solution of the Euler Equations by a Multigrid Method," *Applied Math. & Comp.* 13, 1983, pp. 327-356.
23. Jameson, A., "Multigrid Algorithms for Compressible Flow Calculations," *Proceedings of the 2<sup>nd</sup> European Conference on Multigrid Methods, Lecture Notes in Mathematics*, Volume 1228, Springer-Verlag, 1986, pp. 166-201.
24. Smith, S., Reuther, J., Kinney, D., and Saunders, D., "Low Speed Aerodynamics and Landing Characteristics of Sharp-Class Crew Transfer Vehicle Concepts," AIAA Paper 2001-2888, June 2001.
25. Jespersen, D.C., Pulliam, T.H., and Buning, P.G., "Recent Enhancements to OVERFLOW," AIAA Paper 97-0644, January 1997.
26. Klopfer, G.H., Hung, C.H., Van der Wijngaart, R.F., and Onufer, J.T., "A Diagonalized Diagonal Dominant Alternating Direction Implicit (D3ADI) Scheme and Subiteration Correction," AIAA Paper 98-2824, June 1998.
27. Cooper, G.E., and Harper, R.P., Jr., "The Use of Pilot Rating in the Evaluation of Aircraft Handling Qualities," NASA TN D-5153, April 1969.
28. Zuniga, F. A., Cliff, S E., Kinney, D.J., Hawke, V.M., and Tang, C.Y., "Vehicle Design of a Sharp CTV Concept using a Virtual Flight Rapid Integration Test Environment," AIAA Paper 2002-4881, August 2002.
29. Venkatakrisnan, V., "On the Accuracy of Limiters and Convergence of the Residual Norms," AIAA Paper 93-0880, January 1993.
30. Gill, P.E., Murray, W., Saunders, M.A., Wright, M.H., *User's Guide for NPSOL (Version 4.0). A FORTRAN Package for Nonlinear Programming*, Dept. of Operations Research, TRSOL86-2, Stanford Univ., Stanford, CA, 1986.